RESEARCH MACHINES

LOCAL VIEWDATA SYSTEM

For 380Z Disc Systems

USERS MANUAL

DN 1117

LOCAL VIEWDATA SYSTEM User Manual

Release 1, September 1982, PN 11117

Copyright (C) 1982 by Council for Educational Technology.

All rights reserved. Copies of this publication may be made by customers exclusively for their own use, but otherwise no part of it may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language without the prior written permission of

Council for Educational Technology for the UK
3 Devonshire Street
LONDON W1N 2BA
England.
Telephone London (01) 636 4186

The original labelled distribution disc is regarded as the only proof of purchase and must be produced in order to qualify for an update at a reduced rate. Keep it safe and always work from copies.

Prestel and the Prestel symbol are registered trademarks of British Telecom.

Additional copies of this publication may be ordered from Research Machines at the address below. Please ask for
"LOCAL VIEWDATA SYSTEM User Manual, PN 11117".

Research Machines Ltd.
Post Office Box 75
Mill Street
Oxford OX2 0BW
England.
Telephone Oxford (0865) 249866

LOCAL VIEWDATA SYSTEM USER MANUAL

CONTENTS

Chapter 1	Introduction	1.1
Chapter 2	Getting Started with Local Viewdata	2.1
	2.1 The USER program	2.1
	2.2 Keyboards and keyboards	2.2
	2.3 The database	2.3
	2.4 The EDITOR program	2.4
	2.5 The ENTER command	2.6
	2.6 Short cuts	2.8
Chapter 3	More on Local Viewdata	3.1
	3.1 Introduction	3.1
	3.2 Viewdatabase structure	3.1
	3.3 Routing structures	3.2
	3.4 Typical viewdatabases	3.4
	3.5 Maximum number of frames	3.4
	3.6 Starting a new database	3.4
	3.7 Entering command options	3.5
	3.8 Entering page numbers	3.5
	3.9 The editing commands	3.6
	3.10 The text editor	3.8
	3.11 The routes editor	3.10
	3.12 Absorbing frames from Prestel	3.12
Appendix A	Text Editor Quick Reference Guide	A.1
Appendix B	Error messages	B.1
Appendix C	Machine language subroutines	C.1
Index		I.1

CHAPTER 1

INTRODUCTION

WHAT ARE VIEWDATA SYSTEMS?

Viewdata systems are computer based information retrieval systems accessed via the telephone network. They are characterised by their very simple user interface so that searching for information, although often slow, requires no special training. The information received is normally presented on a modified domestic television receiver, but other devices such as a microcomputer can also be used. Users of viewdata systems need only numeric keypads since information is found by choosing from a series of numbered options presented on their TV screens.

Information is provided in blocks which occupy the whole TV screen. Each block is called a FRAME.

WHAT IS PRESTEL?

Prestel is the name of the public viewdata system which is operated by British Telecom. It is based upon a network of minicomputers which stretches over much of the country so that you can expect to pay only local call charges to your nearest computer. Prestel was the first viewdata system in the world and was a concept originally developed in this country by British Telecom.

Prestel now has thousands of users, hundreds of organisations providing information and hundreds of thousands of frames of information.

THE 380Z LOCAL VIEWDATA SYSTEM

The 380Z Local Viewdata system allows blocks of text and low resolution graphics to be created, which are interconnected by pointers exactly like remote viewdata systems such as Prestel. These blocks of text and their pointers may then be altered or deleted as required. Information from Prestel may be absorbed into the local system using the TERMINAL program described in the PRESTEL TELESOFTWARE USER MANUAL.

HOW TO USE THIS MANUAL

Chapter 2 is written as a tutorial to introduce you to using and editing a viewdata system. By far the best way to learn is to sit down at the keyboard and try it for yourself.

Chapter 3 describes the Local Viewdata System in more detail and tells you how to use the facilities provided to create your own database.

Appendix A is a quick reference guide, listing the most commonly used commands.

Appendix B is a list of error messages generated by the Local Viewdata System.

Appendix C describes the support routines provided in BASICSV to access a remote Prestel database and to setup and manipulate your own local database.

CHAPTER 2

GETTING STARTED WITH LOCAL VIEWDATA

2.1 THE USER PROGRAM

One part of the Local Viewdata System is the facility to search through frames of information stored on disc on your machine. These frames are exactly like frames from a remote viewdata system such as Prestel, and the method of searching for them is identical.

This function is performed by the program "USER". The Local Viewdata System is supplied with a demonstration database to help you learn how to use this program. When working through the examples in this chapter use the demonstration database provided on the distribution disc.

On a full disc 380Z, you must first copy all files on the distribution disc with single letter extensions, (.A, .B, etc), onto a separate disc. This disc will now be referred to as the DEMONSTRATION DATABASE disc. The procedure for copying files is outlined in your Disc System User Guide.

Plug in and switch on your 380Z system as described in your Disc System User Guide. A message similar to:-

```
COS 3.4/M
→
```

should appear on the video screen. If it does not, then press the RESET button on the front panel. The acoustic coupler and cable will not be required.

Insert a copy of the CET/RML TELESOFTWARE DISC in drive A and the DEMONSTRATION DATABASE disc in drive B as described in your Disc System User Guide. When both doors are closed type B on the keyboard. A message similar to:-

```
→B

RESEARCH MACHINES
31K CP/M VERS 1.4B

A>
```

should appear on the screen. If this message does not appear refer to your Disc System User Guide for advice.

Type:-

```
A>BASICSV
```

followed by RETURN. A message similar to:-

```
RML Extended BASIC V 5.0 H
Copyright (C) 1981 by Research Machines

Ready:
```

should appear on the screen. We are now ready to run the USER program supplied as part of the package. Type:-

```
LOAD "USER"
```

followed by RETURN. After a short pause whilst the program is loaded the message:-

```
Ready:
```

will again appear on the screen. Type:-

```
RUN
```

followed by RETURN. The screen will clear to be replaced by the welcome page of the DEMONSTRATION DATABASE. If you have previously been able to use a system like Prestel, you will see that the commands available to you are very similar to those of remote viewdata systems.

If you have not previously used a viewdata system then you should read the following sections carefully.

2.2 KEYBOARDS AND KEYBOARDS

Towards the bottom of the screen in front of you, you will see

```
key 1 to continue
```

The only keys you need to use to search through a database created by the Local Viewdata System are the ten numbers, 0-9, and two special characters, normally called the star character "*" and the square character "#".

Throughout this manual, the characters "*", "#" and "_" will appear, without quotes, as * # and _

The ten numbers are on the top line of the keyboard under the special symbols. The star character * is at the right of the keyboard on the middle row above the colon :. You type this character by pressing down one of the SHIFT keys and while you are holding this key down, press the key with the colon and star on it.

The final character, #, which Prestel calls a square, is more complicated because it is on different places on different keyboards. On older keyboards it is shown as _ and is on the same key as DELT. Just press the DELT key (but not with SHIFT as you would do when deleting characters).

On newer keyboards it is also shown as _ and is on a key of its own to the left of the RETURN key and above the DELT key.

If you have a COS 4.0 machine, the square character will appear on the screen as _ and not as #. Throughout this manual we will represent the square character by #.

Now, with the Welcome page of the demonstration database displayed on the screen type a 1. The screen will clear and be replaced by the next frame of the demonstration database.

2.3 THE DATABASE

The Local Viewdata System database consists of a number of frames (or "screens") of information. These frames are interconnected by pointers routing you from one frame to the next.

Try using the demonstration database. One or more choices are shown on each frame. Type the number corresponding to your choice and the appropriate new frame is displayed on the screen.

Alternatively you can specify a particular page you would like displayed. This is done using the PAGE NUMBER, which is shown on the top right hand corner of the screen, and the special characters star (*) and square (#). For the time being ignore the letter following the page number. Typing:-

* page number #

will display that particular page on the screen. For example:-

*500#

takes you to page 500

If that page does not exist on the database you will get the message:-

NO SUCH PAGE

at the bottom of the screen.

Similarly, because all local viewdata system databases start at page 0

*0#

will take you back to the beginning.

There are two other commands using the special characters.

To display the previous frame on the screen type:

*#

This can be done up to three times in succession, to display the previous three frames on the screen.

If you make a mistake when you type in a page number you can clear the entry by typing:-

*

and starting again.

If you have a printer attached to your 380Z you may like to take printed copies of viewdata frames. Set up the printer (see Extended BASIC Version 5 Reference Manual, Appendix D for details) and type control-P whenever you want to print a copy of the screen.

When you want to leave the USER system type:-

*90#

All these commands are identical to the commands used on Prestel, except for correcting a mistake which is two stars on Prestel, ** instead of a single star *

How to use your 380Z as a Prestel terminal is described in the accompanying PRESTEL TELESOFTWARE USER MANUAL.

2.4 THE EDITOR PROGRAM

The EDITOR program allows you to create your own local databases and to amend and delete their constituent frames. The more advanced features of EDITOR will be dealt with in a later chapter. For the moment, follow this simple example.

Prepare the machine exactly as for the USER program but this time in response to the message:-

```
RML Extended BASIC V 5.0 H
Copyright (C) 1981 by Research Machines

Ready:
```

type:-

LOAD "EDITOR"

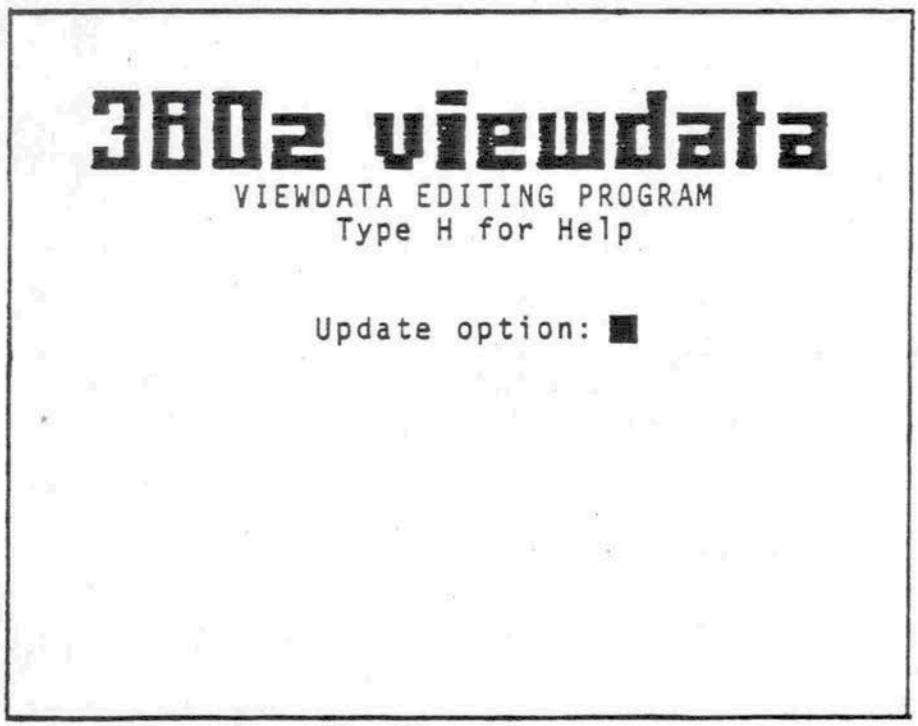
followed by RETURN. After a pause while the program is loaded the message:-

Ready:

will appear again on the screen. Type:-

RUN

followed by RETURN. The screen will clear to be replaced by:-



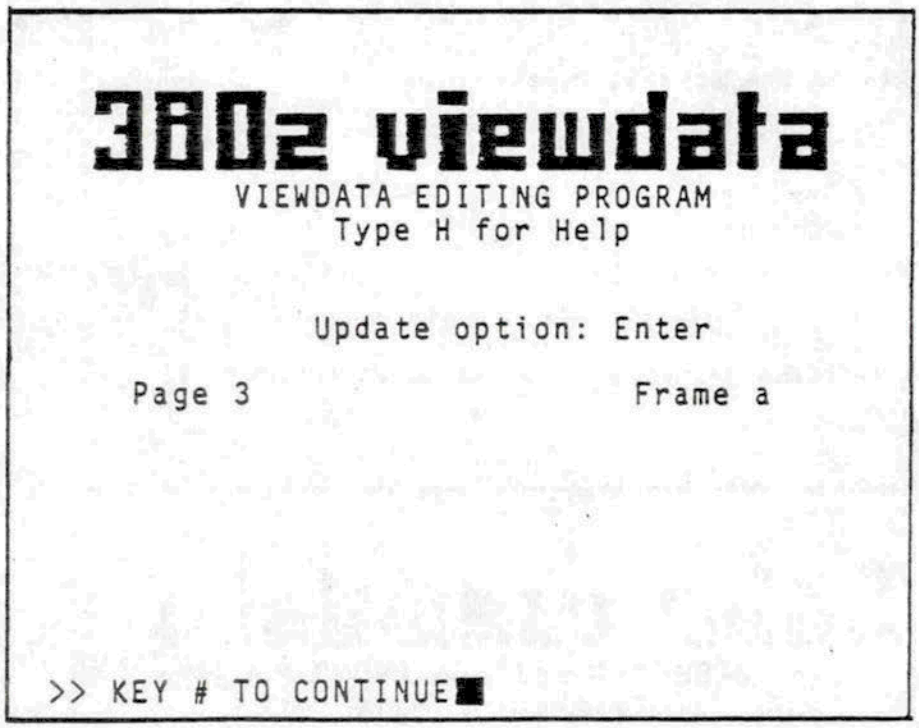
The EDITOR has a command mode structure, that is, commands are entered by choosing from a list of options by typing a single letter (upper or lower case) and after execution of that command the list reappears for the next command letter to be chosen. One of these commands is F which ends execution of the EDITOR program.

If you type the command H the list of available commands is displayed.

2.5 THE ENTER COMMAND

Start off by using the ENTER command. The ENTER command is a request to create a brand new frame for the database. Each frame must have a page number and a letter, called its frame identity.

Type E followed by 3 and #. This means that you wish to create the next frame of page 3. If page 3 has no frames then you will create frame a. After a pause whilst the DEMONSTRATION DATABASE disc is searched the screen will change to:-



The EDITOR will wait for you to type # again to confirm that you wish to create the first frame of page 3. When you do so the screen will become blank except for the top line which will read:-

CET Local Viewdata System 3a

A flashing cursor will appear directly below this top line. You may now type in whatever you wish to be on the frame. There are 22 lines upon which you may write, each of which has 40 character positions. You can move the cursor around the screen using the following cursor control characters:

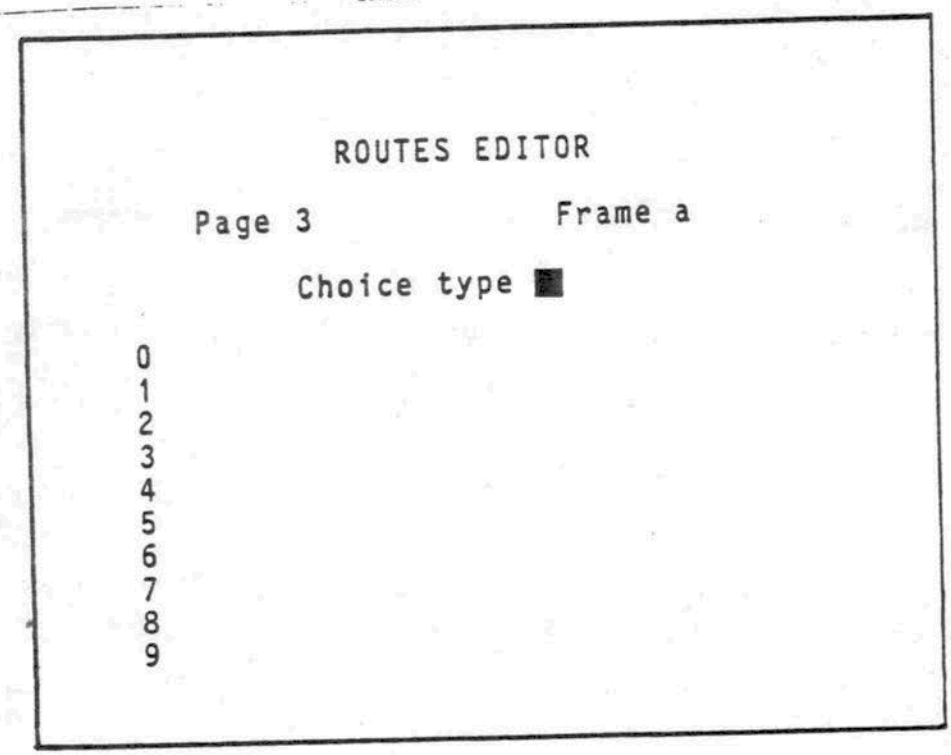
control-J	cursor down
control-K	cursor up
control-H	cursor back
control-X	cursor forward
DELT	cursor backspace delete

Try typing some text, and try using the cursor control characters. When you have finished type:-

Control-Z

This terminates the entering of text on the screen.

The screen is then replaced by one which enables the entering of the routing information for the frame being created:-

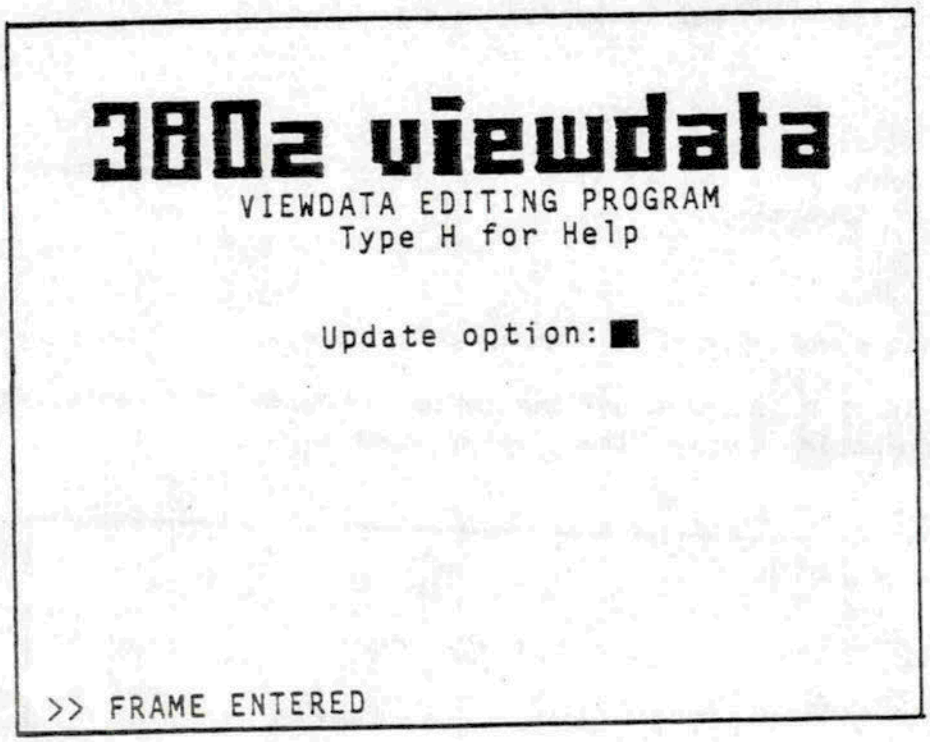


The page number to be accessed for each of the choice digits 0 to 9 must be specified here before the frame is complete. There are various types of routing structure which can be chosen but for now in response to:-

Choice type

just type N meaning NO CHOICES. Asterisks will appear against all choice digits signifying blocked routes and the EDITOR will ask you to confirm this structure.

Type # and after a pause the screen will revert to the list of commands leaving a message on the bottom line of the screen to the effect that the previous command is complete :-



Type F to end execution of the EDITOR program and BASIC will reply:-

```
Ready:
```

If you wish to see the frame just created then load and run the USER program as described on page 2.2 and request page 3 by typing:-

```
*3#
```

2.6 SHORT CUTS

As you become more familiar with loading and running these programs a number of short cuts are desirable.

Firstly, as you may know, loading the BASIC interpreter, loading AND RUNNING a BASIC program may all be done in one command. In response to the message similar to:-

```
RESEARCH MACHINES
31K CP/M VERS 1.4B

A>
```

type:-

```
A>BASICSV USER
```

followed by RETURN. Similarly, if you wish to load BASIC and load and execute the EDITOR program type:-

```
A>BASICSV EDITOR
```

followed by RETURN. Remember to type a space after BASICSV otherwise the error message:-

```
BASICSVEDITOR?

A>
```

will appear.

Secondly, there are fast methods of jumping from the EDITOR program to the USER program and vice-versa. From the EDITOR program one of the commands is U. Typing U (upper or lower case) causes the loading and execution of the USER program. From the USER program typing:-

```
*910#
```

as if requesting page 910, causes the loading and execution of the EDITOR program. Try these out a few times to see their effect.

CHAPTER 3

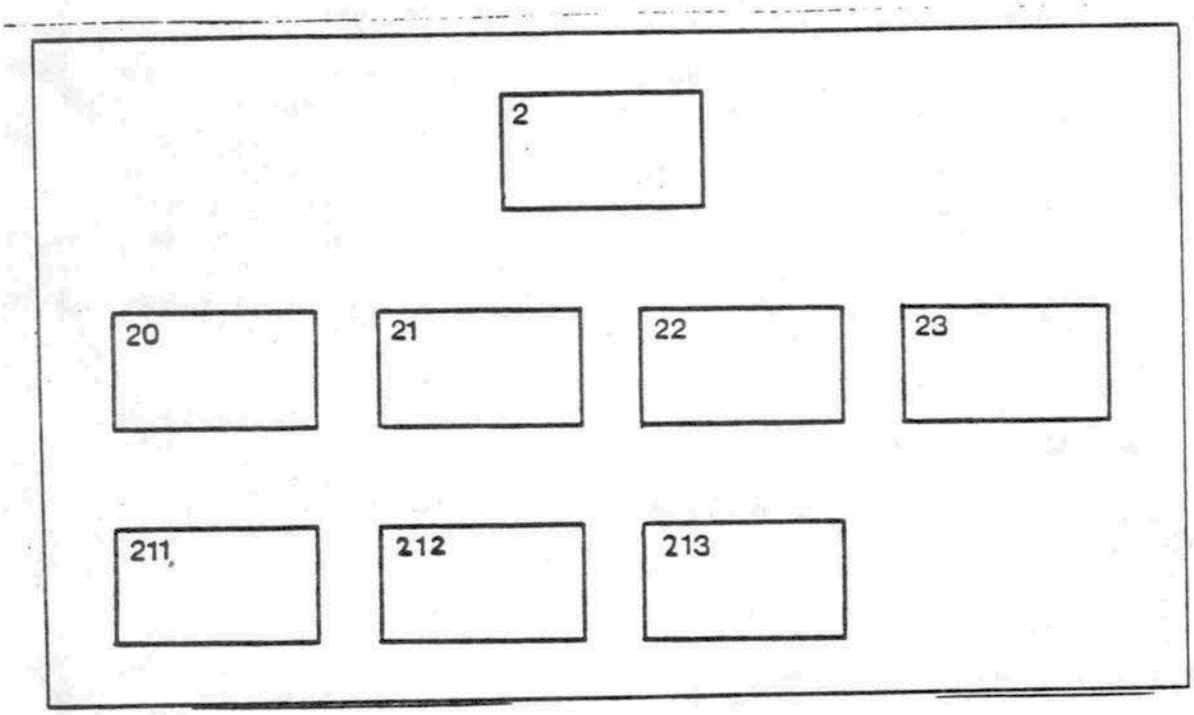
MORE ON LOCAL VIEWDATA

3.1 INTRODUCTION

In this chapter the EDITOR program is described in detail so that you may make maximum use of its facilities.

3.2 VIEWDATABASE STRUCTURE

Data is stored in a viewdata system in the form of PAGES. Each page is identified by a PAGE NUMBER and contains a block of text for display on the screen. There is one special page in the database called the ROOT PAGE given the page number 0. When building a database, page 0 must be created first and then any pages whose page numbers are single digits.



The diagram above shows four pages having two-digit page numbers each beginning with 2. These pages are called FILIALS of page 2. Page 2 is called their PARENT.

Similarly, the pages 211, 212 and 213 are filials of page 21. Page 21 is their parent. All pages beginning with the digit 2 are referred to as DESCENDANTS of page 2. Generally, each page may have up to 10 filial pages whose page numbers are constructed by appending a single digit 0 to 9 to the page number of their parent.

For example, the filials of page 211430 are:-

page 2114300
page 2114301
page 2114302
page 2114303
page 2114304
page 2114305
page 2114306
page 2114307
page 2114308
page 2114309

As you would expect from the analogy, no page may be created until its parent is created and, conversely, no page may be deleted until all its filials have been deleted. This hierarchical relationship, often called a TREE structure, ensures a cohesion which avoids pages being lost (or orphaned, you might say) making database maintenance easier.

Only 24 lines of characters may appear on the 380Z screen at any one time. For the Local Viewdata System the top line is reserved for the page number and the bottom line for system messages leaving 22 lines each of 40 characters available for text. Therefore, pages containing more than 22 lines must be segmented into two or more FRAMES, or "screenfuls" of information. Frames are identified by a letter called the frame identity which is appended to the page number. The first frame of a page is always frame 'a', the second 'b', etc. giving a maximum of 26 frames for any page.

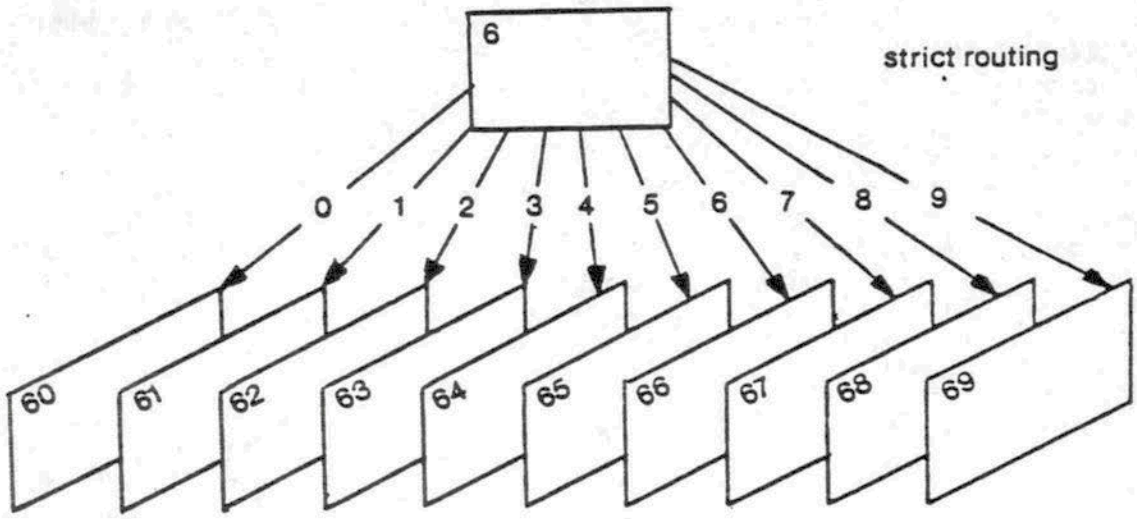
3.3 ROUTING STRUCTURES

If the user, that is the person searching for information in a database, already knows the page number, they can request that page directly by typing * page number # but it is more convenient to guide the user through the database using the routing structures of a viewdata system.

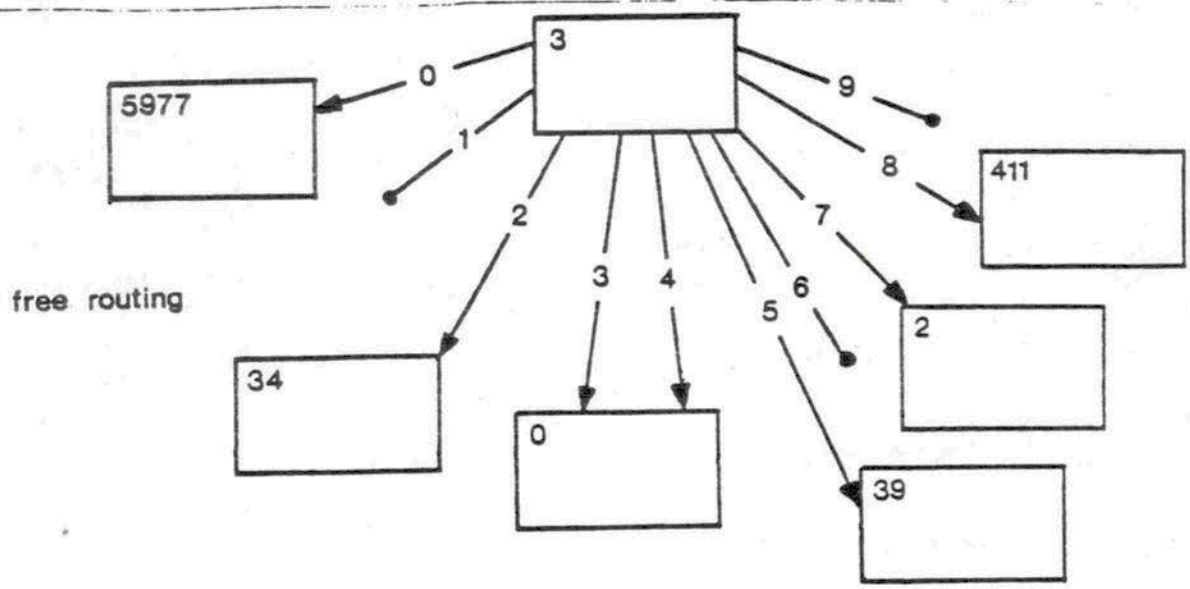
The most commonly used structure is called CHOICE DIGIT ROUTING. Each frame may route the user to up to ten other pages in the database, each of which is accessed by the user keying a single digit, 0-9, while the original frame is displayed on the screen. The appropriate page numbers, corresponding to the single digit routes, are held in a pointer table, which accompanies the text of every frame, but which the user never sees.

When you create a new frame, or alter existing frames, you will need both to enter the text you wish to be displayed on the screen, and also to specify the routing structure you wish to use.

If the 10 routes from a frame refer to its 10 filials in order it is said to exhibit STRICT routing



It is often more convenient to let any route refer to any page, regardless of whether or not it exists at the time, or refer to no page at all. This is called FREE routing.

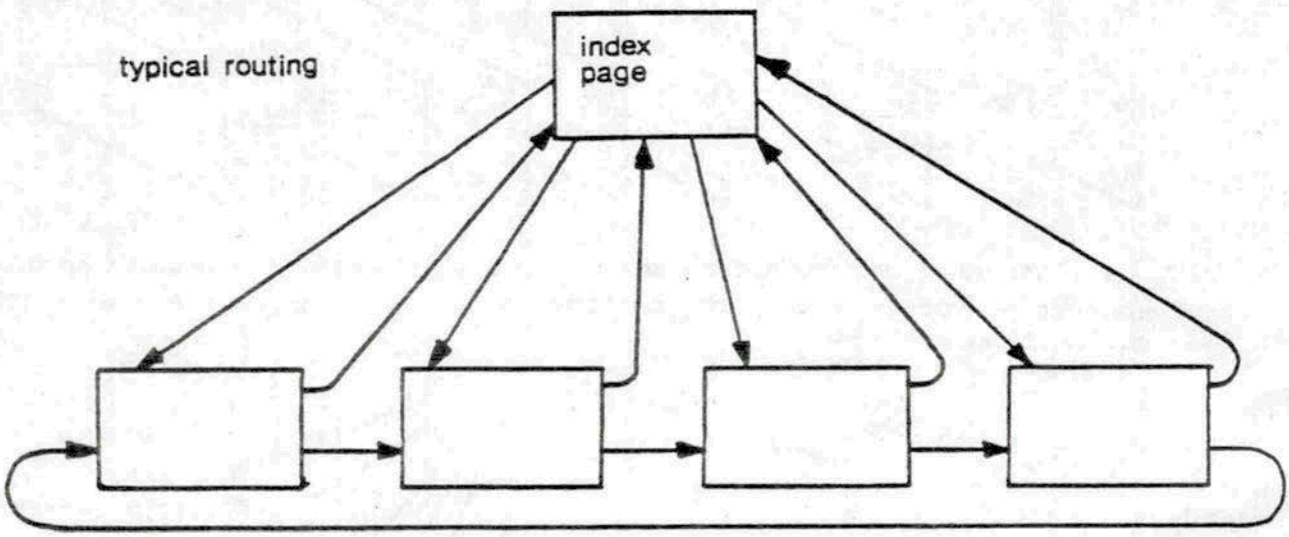


In addition to the single digit routes, 0-9, to any page in the database, you may make use of a # route. This always takes the user to the next frame of that page. For example, if the user types a # from frame 6a, frame 6b will be displayed, a second # will lead to frame 6c and so on.

Note that the single digit routes, 0-9, can only ever route to the "a" frame of a page. Similarly requesting a page directly by typing * page number # will also lead to the "a" frame of that page. The other b-z frames can only be accessed, in alphabetical order, by the # route from the "a" frame.

3.4 TYPICAL VIEWDATABASES

If you are fortunate enough to have access to Prestel, then you can look through the databases of some information providers on Prestel to get some ideas about database design. As a general rule, the page hierarchy is employed to guide users through a sequence of pages containing numbered choices until finally the user arrives at the information required. But for flexibility and user-friendliness other routes are provided in addition to the mainstream filial choices. These allow the user to change direction more easily, to return immediately to some index page, to look at other adjacent items of information and to jump to distant parts of the database which may be of interest.



Of course, a user of Prestel, and the Local Viewdata System, can retrace a path through the last three frames displayed by keying *# but it is more convenient to provide single digit routing.

3.5 MAXIMUM NUMBER OF FRAMES

The maximum number of frames which you are allowed on each disc using this Local Viewdata System is 64.

3.6 STARTING A NEW DATABASE

Follow the connection procedure for your computer system as described in your Disc System User Guide. Place a formatted empty disc in the B drive and the CET/RML TELESOFTWARE DISC in the A drive. Load BASICSV and load and run the EDITOR program as described in Chapter 2.

Your first task is to create page 0 using the ENTER command.

3.7 ENTERING COMMAND OPTIONS

You enter a command option by typing a single letter (upper or lower case) in response to:-

Update option:

It is acted upon immediately and does not require a terminating # or RETURN. The available commands are:-

E	ENTER	Create a new frame
A	ALTER	Alter an existing frame
C	COPY	Create a new frame, which is identical to an existing frame
D	DELETE	Delete one or more frames in a page
T	TREEPRINT	Find all filials of a page
P	PRINTOUT	Print out a frame
U	USER	Load and run the USER program
F	FINISH	End execution of the EDITOR program

If you type any other letter it will be ignored. When a particular command has been executed the message:-

Update option:

will reappear (except in the case of the USER and FINISH commands) inviting you to enter the next command. At any time if you type control-N you will be returned to the command options menu and the current command will be abandoned.

3.8 ENTERING PAGE NUMBERS

Most of the commands above (ENTER, AMEND, DELETE, COPY, PRINTOUT and TREEPRINT) require that you specify a page number. In response to the message:-

Page

which appears beneath the Update option message you may type up to eight digits which make up a page number followed by #. If you make a mistake the DELT key will delete the last character typed.

3.9 THE EDITING COMMANDS

3.9.1 THE ENTER COMMAND

The ENTER command allows the creation of a new frame in the database. Enter the page number of the frame you wish to create as described above. Normally, the EDITOR will respond by identifying the next frame in sequence (if the page does not exist this will be frame 'a') and asking you to confirm by typing #. Typing a character other than # causes a restart of the command.

If you type # you will be presented with a screen which is blank but for the top line which includes the page number and frame identity. You may now enter the contents of the frame using the text editor described below. You must then enter details of the frame's routing structure using the routes editor described below. The frame will then be inserted automatically into the database.

3.9.2 THE ALTER COMMAND

The ALTER command allows the amendment of the text or routing structure of a frame which already exists in the database. Enter the page number of the frame you wish to amend as described above. Normally, the EDITOR will respond by identifying all the frames of that page which exist and asking you to choose one for amendment (upper or lower case letter).

You will be presented with the contents of the frame which you chose. You may now alter the contents of the frame using the text editor described below. You may then alter details of the frame's routing structure if you wish using the routes editor described below. The new copy of the frame will then be inserted automatically into the database replacing the old copy.

3.9.3 THE DELETE COMMAND

The DELETE command allows the deletion of a frame or sequence of frames from the database. Enter the page number of the frame you wish to delete as described above. Normally, the EDITOR will respond by displaying the last frame in sequence of the specified page, asking you to confirm its deletion by typing control-Z. Type control-Z and the frame will be deleted and the next frame in reverse alphabetical sequence similarly displayed for deletion. This continues until either the whole page is deleted or until you type a character other than control-Z terminating the DELETE command.

3.9.4 THE COPY COMMAND

The COPY command allows the creation of a new frame in the database whose contents and routing structure is identical to a frame which already exists. Enter the page number of the frame you wish to copy from as described above. Normally, the EDITOR will respond by identifying all the frames of that page which exist and asking you to choose one for amendment (upper or lower case letter).

Enter the page number of the frame you wish to create as described above. Normally, the EDITOR will respond by identifying the next frame in sequence (if the page does not exist this will be frame 'a') and asking you to confirm by typing #. Typing a character other than # causes a restart of the command.

If you type # you will be presented with a screen whose contents are those of the existing frame but whose page number and frame identity is that of the new frame. You may now alter the contents of the frame if you wish using the text editor described below. You may then alter details of the frame's routing structure using the routes editor described below. The new frame will then be inserted automatically into the database.

3.9.5 THE PRINTOUT COMMAND

If you have a printer attached to your 380Z the PRINTOUT command allows you to take a printed copy of the text AND ROUTING STRUCTURE of a frame which already exists in the database. Enter the page number of the frame you wish to print as described above. Normally, the EDITOR will respond by identifying all the frames of that page which exist and asking you to choose one for printing (upper or lower case letter).

The frame contents and routing structure will then be printed. See your Extended BASIC Version 5 Reference Manual, Appendix D for instructions on setting up a printer.

3.9.6 THE TREEPRINT COMMAND

The TREEPRINT command identifies all filials of a page. Enter the number of the page whose filials you wish to find as described above. Normally, the EDITOR will respond by listing the page numbers of all existing filials of the specified page and the frames which they contain. You are then invited to choose one of the filials so found with which to repeat the exercise or to type # to terminate the TREEPRINT command.

3.9.7 THE HELP COMMAND

The HELP command displays the list of available commands.

3.9.8 THE USER COMMAND

The USER command ends the editing session and immediately loads and runs the Local Viewdata System USER program.

3.9.9 THE FINISH COMMAND

The FINISH command ends the editing session.

3.10 THE TEXT EDITOR

The text editor section of the EDITOR program forms part of the ENTER, ALTER and COPY commands. It allows the insertion and amendment of text and graphics on viewdata frames. A flashing cursor is moved into positions around the screen and characters placed in these positions. You may move the cursor around the screen by using the non-printing characters shown in the diagram below.

Cursor control commands

Control-H	cursor backspace
Control-I	move to next tabulation position
Control-J	cursor down (line feed)
Control-K	cursor up
Control-M	carriage return/line feed
Control-X	cursor forward (non-destructive)
Control-Y	delete to end-of-line
Control-]	cursor home
Control-#	clear screen if keyed twice
DELT	backspace delete

Some of these cursor control characters you may recognise from those of COS 3.4 or COS 4.0. Their implementation here is slightly different in some cases.

The cursor control characters will not allow you to type characters on line 0 and neither will they ever cause the screen to scroll upward. The COS clear-screen character will act only if it is typed twice in succession and leaves line 0 intact. The COS delete-to-end-of-line character is always followed by a carriage-return-line-feed. The COS characters control-A, D, L, O, Q, S, U, V, and W are not implemented.

To exit from the text editor type control-Z. You will then enter the routes editor, (see section 3.11 below). To abandon the whole command type control-N twice and in response to the message:-

>> OPTION CANCELLED Y/N?

type Y to confirm or any other character if you decide to continue with the command.

3.10.1 GRAPHICS

When the 380Z screen is used to display text, it normally contains 24 lines, each of which is up to 40 characters long.

By using the 380Z low resolution facilities, each character position can be further subdivided, once vertically and twice horizontally, into six rectangular units called PIXELS. This is also known as ALPHAMOSAIC graphics, and is the same as is used on most other viewdata systems, including Prestel.

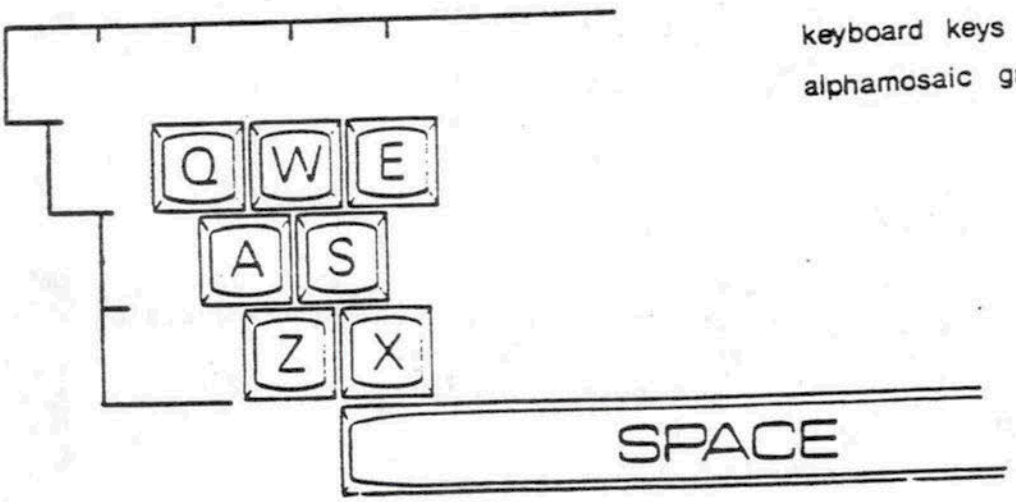
These graphics facilities are incorporated into the Local Viewdata System text editor. Two colours are available, white and grey.

To enter graphics on the screen from the text editor type:-

Control-W if white alphamosaics are required

or Control-G if grey alphamosaics are required

An appropriate message will appear on line 23 and your keyboard will be disabled apart from the cursor control characters and the keys shown below.



keyboard keys for alphamosaic graphics

The six keys Q, W, A, S, Z and X represent the six pixels of the 380Z alphamosaic graphics character. Each of these pixels may be filled, or not, with light so that altogether you can construct 64 different graphics characters in grey and 64 in white.

You must first set up one of these characters by pressing the keys Q, W, A, S, Z and X to switch on the light in the required positions. Pressing any of the keys a second time switches the light off in the relevant position.

As you do this a copy of the character you have constructed will appear on line 23. To enter the graphics character on the screen type:-

E

As you do this a copy of the character will appear on the screen at the cursor position and the cursor will move forward one space. You may then change the character by using the QWASZX keys and typing E again. If you wish to leave a blank space just press the space bar.

To return to ordinary alphanumeric characters type:-

Control-B

The message on line 23 will disappear and you can continue using the text editor.

3.10.2 THE GRAPHICS ALPHABET

You may also type large letters consisting of alphamosaic graphics using the graphics alphabet facility. Press the ESC key and the message:-

) GRAPHICS ALPHABET

will appear on line 23. Any printable character following will appear in alphamosaic graphics three lines high until ESC is pressed again reverting to normal text.

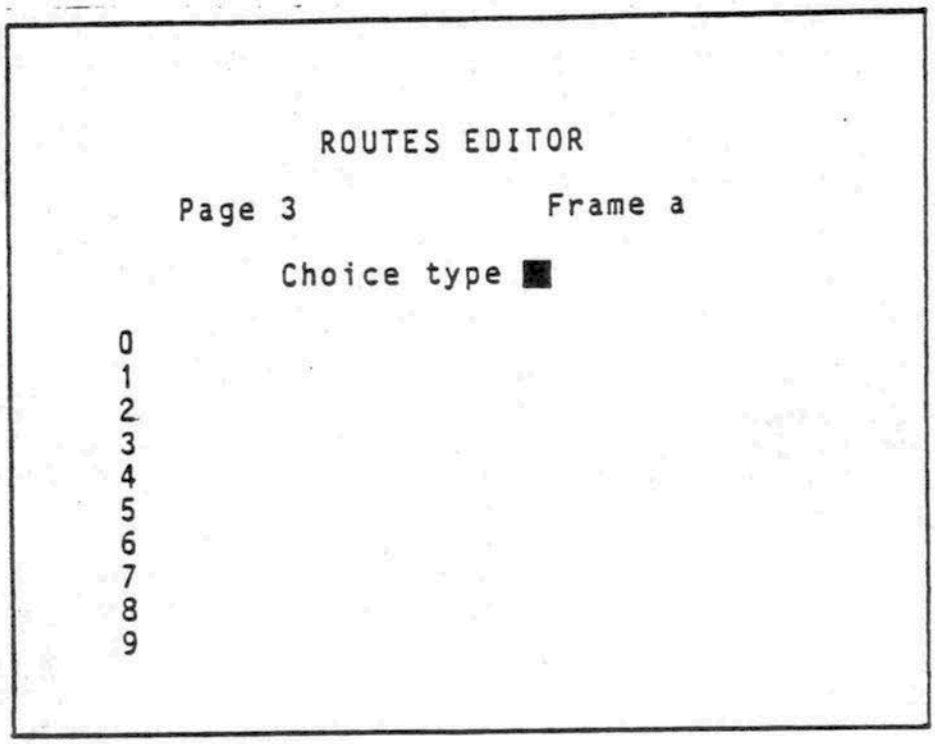
If there are insufficient character positions to accommodate a particular character it will not be accepted from the keyboard. Similarly, if the cursor is incorrectly placed for printing of graphics alphabet characters, ie. the cursor is on line 22 or line 21, then the facility will be disabled. You may type RETURN to start a new line of graphics alphabet characters or DELT to delete characters which you have typed.

3.11 THE ROUTES EDITOR

The routes editor forms part of the ENTER, ALTER and COPY commands. It allows the specification and amendment of the routing structure of a frame. To enter the routes editor from the text editor type:

Control-Z

The screen will clear and be replaced by:



In response to the message:-

Choice type

type one of the following:-

#	All routes remain unchanged
N	No choices. All choice digits are blocked
S	Strict choice. All choice digits 0 to 9 address the 10 filial pages in order. Each filial page number appears in the correct position.
F	Free choice. Enter the referenced page number for each choice at the keyboard terminated with #. If you make a mistake, type DELT to delete the previous digit, or Control-H to return to the previous route. Type * to block a choice digit.
digit	Type 1, 2, 3, 4, 5, 6, 7 or 8 to specify that number of filial routes starting with 0. The numbers of these filial pages will appear in their correct positions. Enter the referenced pages for the unspecified choices as for free choice type above.

When all 10 routes have been specified (either by instruction or explicitly) key # in response to the message:-

>> KEY # TO CONTINUE

or any other character to restart the routes editor.

3.12 ABSORBING FRAMES FROM A REMOTE VIEWDATA SYSTEM

Frames from remote viewdata systems may be saved on disc by the TERMINAL program and accessed via the USER program. These frames are always saved as frames of page 99. Using EDITOR you may COPY these frames to other parts of a database and to other databases, specify their routing structures and alter their text if you wish.

In this way you may absorb individual frames from remote viewdata systems into your own viewdatabases and also reconstruct remote viewdatabases locally.

Appendix A

QUICK REFERENCE GUIDE

This section lists only the most commonly used commands. For full details refer to the relevant chapter.

Load the operating system:

B

Load and run the USER program:

BASICSV USER

followed by RETURN

Load and run the EDITOR program:

BASICSV EDITOR

followed by RETURN

USER program commands

Request a page directly:

*page-number#

Display the previous frame on the screen:

*#

Load and run the EDITOR program:

*910#

Leave the USER program and return to BASIC:

*90#

EDITOR program commands

Cursor control commands

Control-H	cursor backspace
Control-I	move to next tabulation position
Control-J	cursor down (line feed)
Control-K	cursor up
Control-M	carriage return/line feed
Control-X	cursor forward (non destructive)
Control-Y	delete to end-of-line
Control-]	cursor home
Control-#	clear screen if keyed twice
DELT	backspace delete

GRAPHICS commands

For white Alphamosaic graphics:

Control-W

For grey Alphamosaic graphics:

Control-G

To define the graphics character, use the six keys:

q w a s z x

Enter the defined character on the screen:

E

Return to Alphanumeric characters:

Control-B

GRAPHICS ALPHABET commands

For Graphics Alphabet characters:

ESC

To return to Alphanumeric characters:

ESC

Leave the TEXT EDITOR and enter the ROUTES EDITOR:

Control-Z

ROUTES EDITOR commands

#	All routes unchanged
N	No choices. All routes blocked
S	Strict choice
F	Free choice

Appendix B

ERROR MESSAGES

In addition to some of the error messages generated by the BASIC interpreter, the Local Viewdata System will generate messages if you enter page numbers which are invalid for some reason. Note that individual illegal characters, for instance, numeric characters as frame identities, are ignored rather than reported.

>> PAGE HAS NO PARENT

This message is generated by the ENTER and COPY commands when an attempt is made to create a page whose parent has not previously been created.

>> PAGE DOES NOT EXIST

This message is generated by the ALTER, DELETE, COPY, PRINTOUT and TREEPRINT commands when the page specified does not exist in the database.

>> PAGE FULL

This message is generated by the ENTER and COPY commands when an attempt is made to create a new frame for a page which already has 26 frames.

>> PAGE HAS FILIALS - frame 'a' remains

This is a warning message generated by the DELETE command when an attempt is made to delete entirely a page which has filials.

Appendix C

MACHINE LANGUAGE SUBROUTINES

INTRODUCTION

The file BASICSV.COM, which forms part of this package, is an RML Extended BASIC Version 5 interpreter, to which several machine code subroutines have been added. These subroutines perform functions which are not available in BASIC Version 5. They are used extensively within the TERMINAL, USER and EDITOR programs.

Please note that documentation of these subroutines is included for completeness only. They may be withdrawn or modified at any time, and are unlikely to be available in future releases.

ERROR MESSAGES

There are two error messages which can be generated by the routines described below. These are:

```
Routine XXXX called with wrong number
of arguments
```

and

```
Routine XXXX called with bad argument
```

where XXXX is the name of the routine which has been called.

If these errors occur then you will see one of the above messages and then

```
Recovered
Ready:
```

At this point the program that was running has stopped executing, but your program should still be intact. You can correct the error and then rerun the program.

In addition the routines FORTY and ATTACH may produce the message

```
Sorry - requires COS 3.4 or later
```

if you are running on a system which has a version of COS earlier than COS 3.4. The routines will attempt to return to CP/M via a JP 0000H instruction.

ROUTINES

The following routines are described individually below:

HELP	Display help message or version number
SVLN	Save a line from the screen
FORTY	Check version of COS and select 40 character screen
CURPOS	Return the x,y position of the COS cursor
ATTACH	Check verson of COS, select 40 character screen and log on to Prestel system
USER	Emulate Prestel terminal
SEND	Send a character to Prestel
LISTEN	Wait for and display frame from Prestel
STATUS	Return status of preceding command

Routine HELP

This routine is provided to give the user information about the subroutines described in this appendix, and to display the version number of BASICSV which is being used. In addition, a further call to this routine will display the version numbers of all the related programs which make up this package. This allows you to check that you are running the correct versions of the programs.

Method of use:

a) to display a list of example calls to these routines

```
CALL "HELP"
```

or

```
CALL "HELP",0
```

b) to display the version number of BASICSV that you are running on the screen

```
CALL "HELP",1
```

c) to display the version number of BASICSV and the version numbers of the related programs

```
CALL "HELP",2
```

Routine SVLN

The routine SVLN allows the user to read the contents of the screen, line by line, into a string variable. Only low resolution graphics and text are returned to the user; any high resolution graphics are ignored.

Because of the way that Extended BASIC Version 5 optimises the storage of string variables, there is a great danger that you might corrupt the program that you are running if you do not follow the instructions given below for setting up the string variable and calling the routine. Failure to do this will typically result in several of your program lines, possibly completely unrelated to the call to SVLN, being overwritten.

Method of use:

```
CALL "SVLN",N,VARADR(L$)
```

where

N is the line number of the line of the screen that you wish to read; the lines are numbered from 0 to 23, 0 being the top line and 23 the bottom line. Calls to SVLN with N outside this range will result in the routine returning immediately to the caller; L$ will be unchanged and no error message will be issued.

L$ is a string variable which has previously been initialised as shown below.

Initialising the string variable L$

In order to ensure that corruption of your program does not occur when you use this routine, it is recommended that you always:

a) at or near the start of any program that will call SVLN, define a string variable consisting of forty spaces in the manner shown below:

```
N$=""
FOR X=1 TO 40
N$=N$+" "
NEXT X
```

b) before every call to SVLN, set L$ equal to N$, ie. you are using N$ as a string constant set to forty spaces to initialise L$ before using L$ in the call.

Thus each call to SVLN would look like this:

```
L$=N$
CALL "SVLN",N,VARADR(L$)
```

Although this may look an extremely long winded and tedious way of setting up the string variable and the call to SVLN, you should be aware that failure to comply with this procedure will probably result in corruption of the program you are running. YOU HAVE BEEN WARNED !

One further point is that you may initialise L$ to either less than, or greater than forty spaces. If L$ is initialised to more than forty spaces,then the routine will return the whole line in the first forty positions in the string variable and will leave the remainder unchanged.

If, however, the string variable is less than forty characters in length, it will return only the first p characters of the line, where p is the length of the string variable.

Character conversions

The routine does not return an exact copy of the screen; it makes three character conversions. These are:

a) characters which are read from the screen as nulls (00 hex) are converted by the routine to the character O (4F hex)

b) characters which are read from the screen as either of the graphics blank characters (80 hex or C0 hex) are converted by the routine to space characters (20 hex)

Example of use:

The program TERMINAL.BAS contains several examples of the use of SVLN.

Routine FORTY

This routine is used to select a forty character screen width on those systems which have a 40/80 character VDU board. On other systems it has no effect. It also checks the version of COS on which the program is running and if this is not COS 3.4 or a later version of COS, will output the message

Sorry - requires COS 3.4 or later

An attempt will then be made to return to the operating system CP/M via a JP 0000H instruction.

Method of use:

CALL "FORTY"

Example of use:

The programs USER.BAS and EDITOR48.BAS both call this routine within the first few lines of the program.

Routine CURPOS

This routine allows a BASIC programmer to find out where on the screen the cursor is currently positioned. The routine returns two values which are the x and y co-ordinates of the cursor position.

The horizontal co-ordinate x is in the range 0 to 39, and the vertical co-ordinate y is in the range 0 to 23. The origin (0,0) is positioned at the top left hand corner of the screen.

Method of use:

```
CALL "CURPOS",VARADR(X),VARADR(Y)
```

The x co-ordinate of the cursor position is returned in the variable X, the y co-ordinate is returned in the variable Y.

Example of use:

This routine is used in both of the editing programs EDITOR31.BAS and EDITOR48.BAS.

VIEWDATA specific routines

The remaining routines described here are of interest only to those people who wish to write their own programs which will access a viewdata system (such as PRESTEL) using an acoustic coupler or a modem.

The routines ATTACH, USER and LISTEN can return status values which give information about the functions they have performed. In order to obtain this information, the user must first call the appropriate routine and then make a call to the routine STATUS. The routine STATUS returns a value which reflects the result of the previous routine's action. The values which can be returned are described in the section which relates to the routine which produces this value.

Routine ATTACH

This routine is used to set up the link between the 380Z and the PRESTEL system (or another viewdata system). Once the routine has been called the 380Z is initialised ready to logon to the PRESTEL system and instructions to the user are displayed on the screen.

If you wish to use this package with a viewdata system which does not use the same logon procedure as PRESTEL, you may have to force the program to exit from the ATTACH routine by typing any key on the keyboard (see below). You can then follow the logon procedure for the system you are using.

Method of use:

```
CALL "ATTACH"
```

A screen of instructions is displayed.

Example of use:

There is an example of the use of ATTACH in the terminal program, TERMINAL.BAS.

Status returns:

The routine returns one of two values via the STATUS routine (see below).

If the logon has proceeded normally and the program has received an ENQ character (the send me your identification number prompt) from the PRESTEL system and has transmitted its identification number, then the status value returned is 4.

If, however, the user interrupts the ATTACH routine by typing any key on the keyboard, then the routine returns immediately and the status value is set to 3.

Further details of the routine.

Because this routine has been written to allow you to use either PRESTEL or some other viewdata system, and also allows you to use either an acoustic coupler or a modem, there are several features of the routine that some users may wish to have more details of. The following is an extract from the program documentation:

```
      *H Routine ATTACH

      ;
      ; Initialise and log on
      ; ---------------------

      ;
      ; This routine is called by CALL "ATTACH"
      ;
      ; On entry the C register contains the number of arguments
      ; and should be zero.
      ;
      ; The routine performs several initialisation functions:-
      ;
      ;  The SIO4 is initialised and the data rate for receive
      ;  is selected.
      ;
      ;  The version of COS on which we are running is obtained
      ;  via a call to routine SETVER and if it is COS 4.0 or
      ;  later, then a call to EMT WIDTH to select a 40 character
      ;  screen is made.
      ;
      ;  We then clear the screen, output the version number and
      ;  the disc type on the top line and a front page after that.
      ;
      ;  The routine then loops for 20 seconds waiting either for
      ;  an ENQ from Prestel, or for the user to type any key on
      ;  the keyboard.
      ;
      ;  If the 20 second timer expires then we transmit a '*' to
      ;  rerequest the logon frame and go back to the start of the
      ;  wait loop.
      ;
      ;  After receipt of an ENQ, the routine transmits the terminal
      ;  I/D and password, clears the screen and returns to caller.
      ;
      ;  The wait loop and transmission of the terminal I/D can be
      ;  abandonned at any time by typing any key on the keyboard.
      ;  In this case, the routine just returns to caller (without
      ;  clearing the screen).
```

Routine USER

This routine allows you to use your 380Z as a normal Prestel terminal much as any other Prestel user would use a Prestel set or adaptor. Characters sent by Prestel are displayed on the 380Z screen, and characters typed on the keyboard of the 380Z are sent to Prestel.

This function will continue until you type one of the following four keys on the keyboard:

CTRL/C	Control and C
CTRL/P	Control and P
CTRL/S	Control and S
CTRL/T	Control and T

When any of these keys are typed on the 380Z keyboard, the routine USER exits leaving a status value which can be read by the routine STATUS.

Several of the normal functions of a Prestel terminal can not be implemented easily on the 380Z, so you will find that the display on your 380Z will not correspond exactly to that seen on a normal Prestel terminal. In particular, colours on Prestel can only be shown on the 380Z as white or grey, and facilities such as double height and flashing characters are not available.

Method of use:

```
CALL "USER"
```

The routine will return to the caller when one of the following characters is typed on the keyboard: CTRL/C, CTRL/P, CTRL/S or CTRL/T.

Example of use:

You will find an example of the use of this routine at the beginning of the terminal program, TERMINAL.BAS.

Status returns:

The routine will return a value via the STATUS routine corresponding to the particular key of the four exit keys which were typed. These values and the corresponding keys are:

Status value	Key typed
0	CTRL/C
1	CTRL/P
2	CTRL/S
3	CTRL/T

Routine SEND

This routine is provided to allow you to send a single character to the Prestel system from a BASIC program. You are not restricted to simply sending those characters which are read from the keyboard as in the USER routine described above.

Method of use:

```
CALL "SEND",ASC(C$)
```

where

C$ is the character you wish to send

The routine will send the character to the Prestel system and then return to the caller.

If you want to send several characters then you must make several separate calls to this routine. Only one character will be sent for each call.

Example of use:

This routine is used in the last section of the terminal program, TERMINAL.BAS, and you will see that a call to SEND is usually followed by a call to the routine LISTEN which is described below.

Routine LISTEN

This routine waits for characters from the Prestel system and displays them on the screen as they are received. If no characters are received during the first five seconds then the routine returns to the caller. Once a character has been received, the routine will wait for half a second after the last character before returning to the caller. Status values are returned to the caller to allow him to determine whether any characters were received and also whether there were any parity errors (as a result of noise on the telephone line) in the received characters.

Method of use:

```
CALL "LISTEN"
```

The routine returns to the caller either after five seconds if no characters were recieved, or half a second after the receipt of the last character.

Example of use:

This routine is used in the terminal program, TERMINAL.BAS, usually adjacent to the calls to SEND. Note however that the terminal program does not make use of the status values returned by this routine (see below for details).

By using the routines SEND and LISTEN together, it is quite easy to write a program in BASIC which will select and display a given Prestel page.

The following extract from an example program will request page number 2114304 and display it on the screen (assuming you are already logged on to the remote system) :-

```
100 LET PAGE$="2114304"

110 CALL "SEND",ASC("*")

120 FOR X=1 TO LEN(PAGE$)
130 CALL "SEND",ASC(MID$(PAGE$,X,1))
140 NEXT X

150 CALL "SEND",ASC("#")

160 CALL "LISTEN"
```

Status returns:

The routine will return a status value via the routine STATUS to indicate the following three situations:

Status value	Meaning
0	A frame has been received and contains no transmission errors
1	No characters were received during a period of five seconds
2	The frame received contains parity or other transmission errors

Routine STATUS

This routine is used to get the status value which has been produced by one of the routines described above. You should use it immediately after the routine which generates the status value you wish to read as the other routines will overwrite this value.

The particular significance of the status values returned are described above in the section relating to the routine which generates the value.

Method of use:

```
CALL "STATUS",VARADR(ST)
```

where

ST is the status value returned by the routine previously called

Example of use:

An example of the use of the STATUS routine can be found in the terminal program, TERMINAL.BAS, where it is used in several places e.g. after the call to ATTACH, and shortly afterwards after the call to USER.

Local Viewdata System User Manual

INDEX

Acoustic coupler	2.1
Alphamosaic graphics	3.9
ALTER	3.6
ATTACH	C.5
Backspace delete	3.8
BASICSV	2.2,2.9
Carriage return	3.8
Choice digit	3.2
Clear entry	2.4
Clear screen	3.8
Command mode structure	2.5,3.5
Control characters	
Control-B	3.10
-G	3.11
-H	3.10
-J	3.10
-K	3.10
-M	3.10
-N	3.5
-P	2.4
-W	3.11
-X	3.10
-Y	3.10
-Z	3.10
-]	3.10
-#	3.10
COPY	3.7
COS 4.0	2.3,3.8
CURPOS	C.4
Cursor backspace	3.8
Cursor control commands	2.7,3.8
Cursor down	3.8
Cursor forward	3.8
Cursor home	3.8
Cursor up	3.8
Database structure	3.1
DELETE	3.6,3.8
Delete-to-end-of-line	3.8
DELT	2.3,3.5,3.8,3.10
Demonstration database	2.1,2.2
Descendant	3.1
EDITOR program	2.4,2.9
ENTER	2.6,3.6

Error messages B.1
>> PAGE HAS NO PARENT
>> PAGE DOES NOT EXIST
>> PAGE FULL
>> PAGE HAS FILIALS - frame 'a' remains

ESC 3.10

Filial 3.1,B.1
FINISH 3.8
Flashing cursor 2.6
FORTY C.4
Frame 1.1,3.2,3.4
FREE routing 3.3

Graphics 3.9
Graphics alphabet 3.10
Graphics characters 3.10

H 2.5
HELP 3.7,C.2
Hierarchical relationship 3.2

Keyboards 2.2

Line feed 3.8
LISTEN C.8

Machine code subroutines C.1

New database 3.4
NO CHOICES 2.7
NO SUCH PAGE 2.3
Number of frames 3.4

Page number 2.3,3.1,3.5
PAGES 3.1
Parent 3.1
Pixel 3.9
Previous frame 2.4
Printing frames 2.4,3.7
PRINTOUT 3.7

Quick reference guide A.1

RETURN 3.8,3.10
Root page 3.1,2.4
Routes editor 2.7,3.10
Routing structure 2.7,3.2